Notes on a sensor model for AURORA

Gabriel Gomes

1 Preamble

- A model of a loop detector is a requirement for simulating signal control. For example, the so-called *actuated* intersection controller extends the green light indication when it receives a *call* from an approaching vehicle.
- Discrete vehicle events are also used in coordinated and adaptive signal control algorithms.
- A useful sensor model must translate the macroscopic variables into discrete vehicle passage events.
- At the software level, we want a general design that may later serve as a basis for a variety of sensor types, including probe vehicles.

2 Code structure

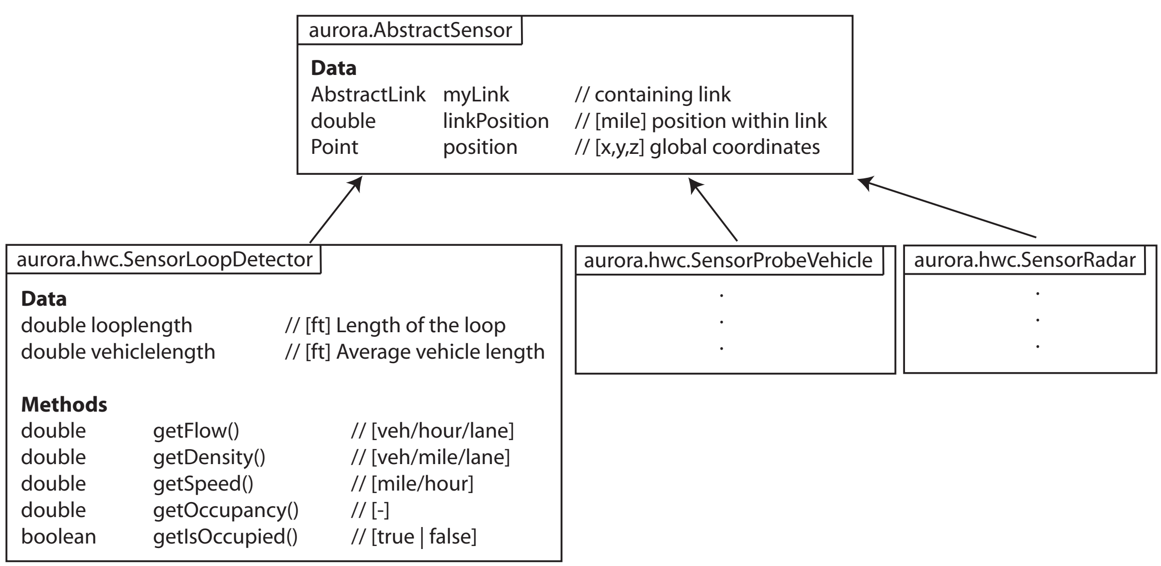

Figure 1: Program objects

3 Loop detector model

3.1 Notation

Dimensions:

- $L \ldots$ Length of the link [mile]
- $L_s \ldots$ Length of the loop detector [ft]

Macroscopic link variables:

- $f_{in} \ldots$ flow entering link [veh/hour/lane]
- $f_{out} \ldots$ flow exiting link [veh/hour/lane]
- $\rho \ldots$ density [veh/mile/lane]
- $v \ldots$ speed [mile/hour]

Sensor variables:

- $f_s \ldots$ flow [veh/hour/lane]
- $\rho_s \ldots$ density [veh/mile/lane]
- $v_s \ldots$ speed [mile/hour]
- $o_s \ldots$ occupancy [-]

3.2 Flow, Density, Speed, Occupancy

Density is assumed spatially uniform across the link:

$$\rho_s = \rho \tag{1}$$

The uniform density profile is assumed to vary linearly throughout the simulation time interval, from $\rho(x,t)$ to $\rho(x,t+t_s)$. From the conservation of vehicles $\rho_t + f_x = 0$ we get that flow varies linearly in space, so a sensor placed at position x within a link of length L will register

$$f_s = \frac{L-x}{L} f_{in} \; + \; \frac{x}{L} f_{out} \tag{2}$$

We can compute speed in one of two ways. The first is the usual definition,

$$v_s = f_s/\rho_s \tag{3}$$

This formula does not necessarily yield $v_s = v_f$ for sub-critical densities because it does not account for onramp and offramp terms affecting the flow. We may need a more complicated formula to ensure $v_s(\rho_s < \rho_{crit}) = v_f$. For the ACTM, such a formula looks like:

$$v = \frac{f}{(n + \gamma r)(1 - \beta)} \frac{L}{\Delta t} \tag{4}$$

An analogous equation is needed for AURORA. For the occupancy there are similarly two possible formulas. The straightforward dimensional formula is:

$$o_s = \rho \times L_s / 5280 \tag{5}$$

$\rho \times L_s$ is the feet covered by vehicles in an average mile and an average lane. Dividing this by the total feet in a mile gives the occupancy. But this does not ensure $o_s(\rho = \rho_{jam}) = 1$, unless $\rho_{jam} = 5280/L_s$. We can alternatively use

$$o_s = \frac{\rho}{\rho_{jam}} \tag{6}$$

3.3 Is occupied

The function `getIsOccupied(Δt)` returns `true` if the detector has registered a call during the previous Δt seconds, and `false` otherwise.

Assumption: The state corresponding to the time interval $[t - \Delta t, t]$ equals the current state (at time t). This is an obviously invalid assumption when $\Delta t > t_s$ (the simulation time step). It is also invalid within t_s since we have assumed that density varies linearly in time. The assumption is needed unless we record data histories, or we are willing to make "isoccupied" a state of the detector (perhaps not a bad idea).

Assumption: The probability that a vehicle triggers a detector is a Poisson distribution:

$$p(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{7}$$

where k is the number of observed events (i.e. vehicle passages), λ is the expected number of events in the given time interval. The probability of registering at least one event is then:

$$1 - f(0, \lambda) = 1 - e^{-\lambda} \tag{8}$$

Calculation of λ.

At first glance it may seem that λ should simply be the flow. This fails however when vehicles are queued since the flow is then low, but the probability of vehicle actuation should be high. This is especially important for actuated control.

Let us consider the following formula for λ:

$$\lambda = \frac{v\Delta t + L_v}{h + L_v} \tag{9}$$

where L_v is the average vehicle length, Δt is the observation time interval, and h is the mean vehicle distance headway. This formula results from calculating the expected number of vehicle passages on a road with evenly spaced vehicles moving at a constant speed v, as shown below.

Figure 2: Δt=0

With $\Delta t = 0$, the expected number of events equals the vehicle occupancy:

$$\lambda(\Delta t = 0) = \frac{L_v}{h + L_v} \tag{10}$$

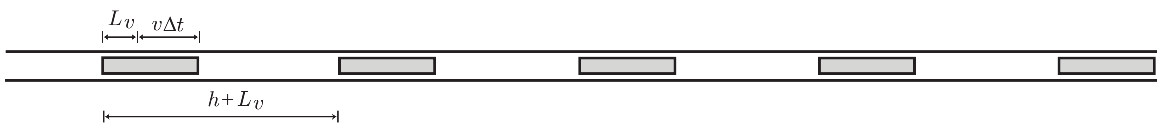

Figure 3: $v\Delta t < h$

Allowing vehicles to move, we trace their trajectory through the time interval Δt, which is assumed in Figure 3 to be less than h/v. The estimate of λ is now the *smeared* occupancy:

$$\lambda = \frac{v\Delta t + L_v}{h + L_v} \tag{11}$$

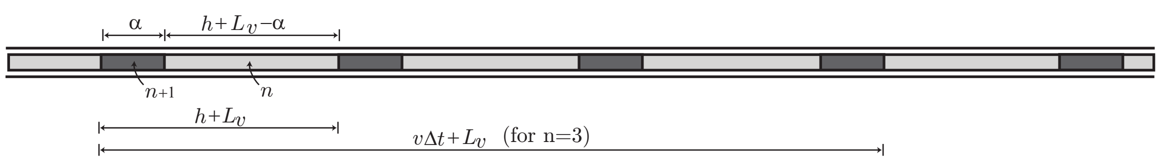

Figure 4: General case

Now, for the general case, allowing $v\Delta t$ to reach arbitrarily large values, we end up with the situation depicted in figure 4 where some reqions have been covered n times, and others $n + 1$ times. n is the integer part of the ratio of the distanced traveled to the inter-vehicle distance.

$$n = \text{floor}\left(\frac{v\Delta t + L_v}{h + L_v}\right) \tag{12}$$

Defining α as the portion covered $n + 1$ times:

$$\alpha = L_v + v\Delta t - n(h + L_v) \tag{13}$$

we can now calculate λ:

$$\lambda = \frac{(n+1)\alpha + n(h + L_v - \alpha)}{h + L_v} \tag{14}$$

$$= \frac{n(h + L_v) + \alpha}{h + L_v} \tag{15}$$

$$= \frac{n(h + L_v) + L_v + v\Delta t - n(h + L_v)}{h + L_v} \tag{16}$$

$$= \frac{v\Delta t + L_v}{h + L_v} \tag{17}$$

Note 1. Because headway is not a macroscopic quantity, the formula actually implemented should be

$$\lambda = (v\Delta t + L_v)\rho \tag{18}$$

where $\rho = 1/(h+L_v)$ is the link density. The average vehicle length is a constant parameter.

Note 2. The assumption of zero detector length can be relaxed. For a detector with length L_s the formula for λ becomes:

$$\lambda \; = \; \frac{v\Delta t + L_v + L_s}{h + L_v} \; = \; (v\Delta t + L_v + L_s)\rho \tag{19}$$

Note 3. These formulas do not account for the fact that consecutive calls to `getIsOccupied()` are not independent. It is more likely that the result will be `true` if a very recent call was `true`, since the vehicle will take some time to clear the detector. This may be resolved easily if we make isoccupied a state of the detector.